Microsoft

.net developer training

Create XML Web Services and Applications with Visual Studio .NET

Code Instrumentation and Middle Tier Technologies

Module 10 Agenda

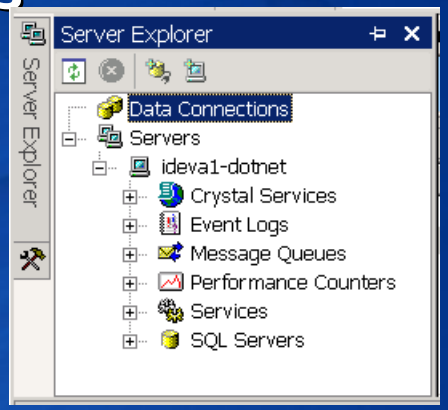
- Application Goals
- Visual Studio .NET Server Explorer
- .NET Messaging, MSMQ
- Windows Services
- COM+ Integration
- Application Diagnostics
  - Performance monitor
  - Event log
  - Tracing

Application Goals

- Scalable
- Reliable
- Tunable
- Testable

Visual Studio .NET Server Explorer

- Server Explorer
  - Tool for working with server features
  - Useful in integrating features into your application
    - SQL Server Data/Schema
    - Event logging
    - Performance monitor counters
    - Windows services
    - MSMQ integration
  - Also useful for exploring server features



.NET Messaging (MSMQ)

- Scalable, transacted, reliable cross-machine and cross-platform messaging
  - Good for high performance applications
- Based on Microsoft Message Queue 2.0
- Send and receive objects, primitives, custom
- Disconnected and connected
- Message Queue administration

Adding MSMQ Support

- Messaging support is baked into Visual Studio .NET
- Queues are accessible from Server Explorer
  - Drag and drop features work with designer
    - Objects are created under the covers
    - Require minimal code from developer:
      - ie. MessageQueue1.Send(OrderInfo)
    - Objects can be serialized into queues

Windows Services

- Unique application model
  - Application that runs in the background with its own
    - credentials
    - error handling
- Many useful scenarios:
  - SQL Server
  - Antivirus
  - IIS – Web server and FTP server
- Managed with Service Control Manager

Windows Services in .NET

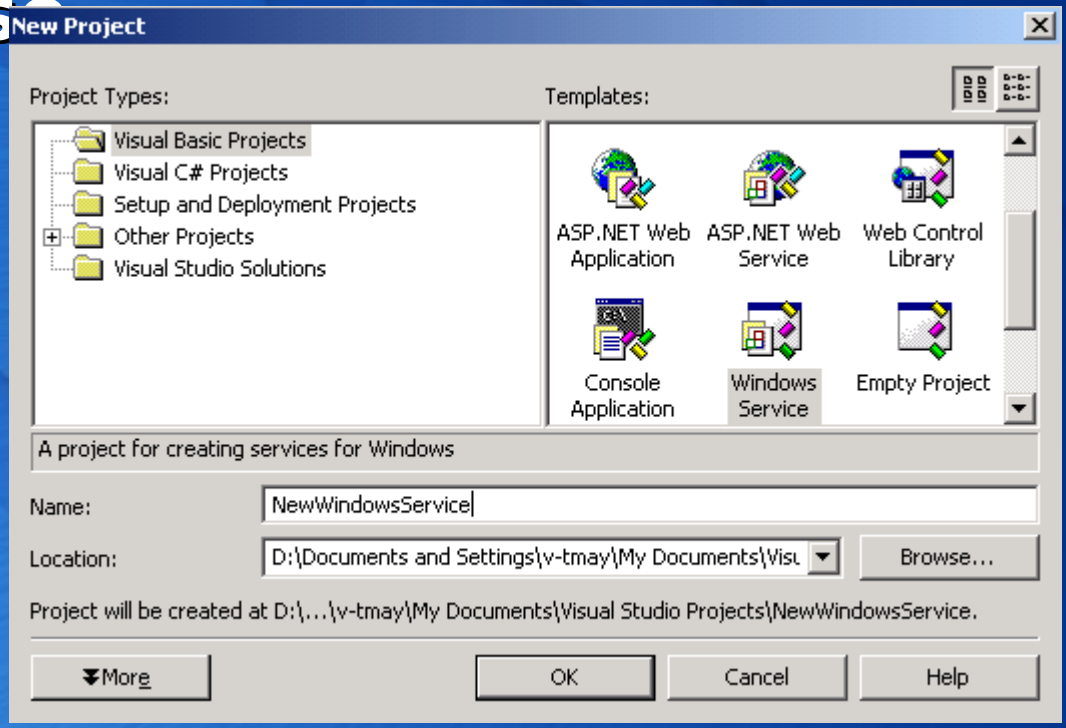
- Built into .NET
  - Extend a service base class
  - Override methods (OnStart, OnStop)

- Respond to events
  - Receive Message Queue
  - Change Directory/file
  - Write event log
  - Timer tick

- General Tasks
  - Perform long-running tasks
  - Provide utility functions
  - Serve as a data transfer engine

Windows Services
Demo

- Windows Service
- Add Windows service to watch for an XML file in a folder, then add the file to the database

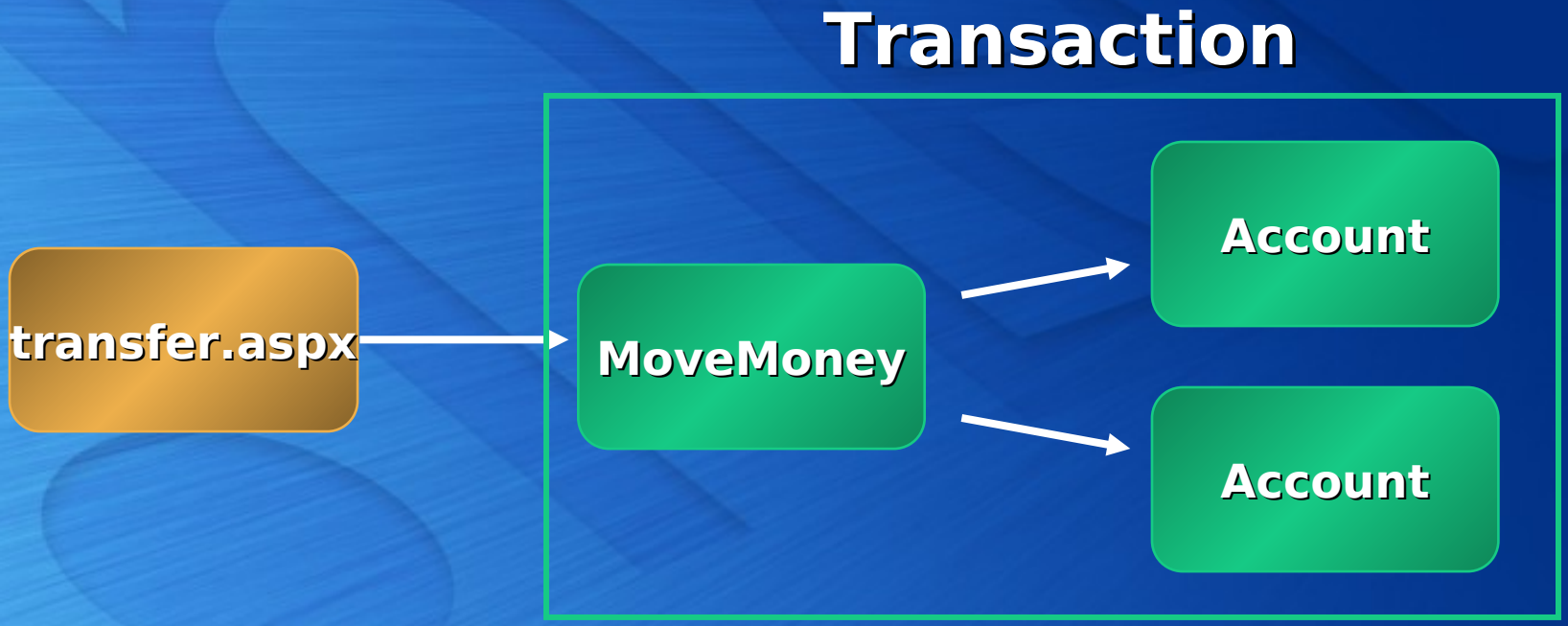


COM+ and .NET Applications

- Attributes define COM+ properties
  - Custom attribute is set in code
  - Compiled into the metadata
- Admin can override some service settings, e.g.
  - Object Pool minimum, maximum, etc
- COM+ Services, CLR and .NET Framework work together

ASP.NET and COM+

- ASP.NET Interacts with COM+
  - Set by transaction directive
  - Interacts with .NET components



Deployment of COM+ Applications

- XCOPY deployment with dynamic registration
  - When component is first used, .NET takes care of
    - Creating COM+ Applications
    - Adding .NET Components to COM+ Applications
    - Configuring services in COM+ catalog
- Preload
  - Create COM+ Application - COM+ Explorer, Script
  - Add Assembly to the GAC (Global Assembly Cache) - Gacutil.exe or Drag and Drop
  - Add Assembly to COM+ Application -

Application Diagnostics Topics

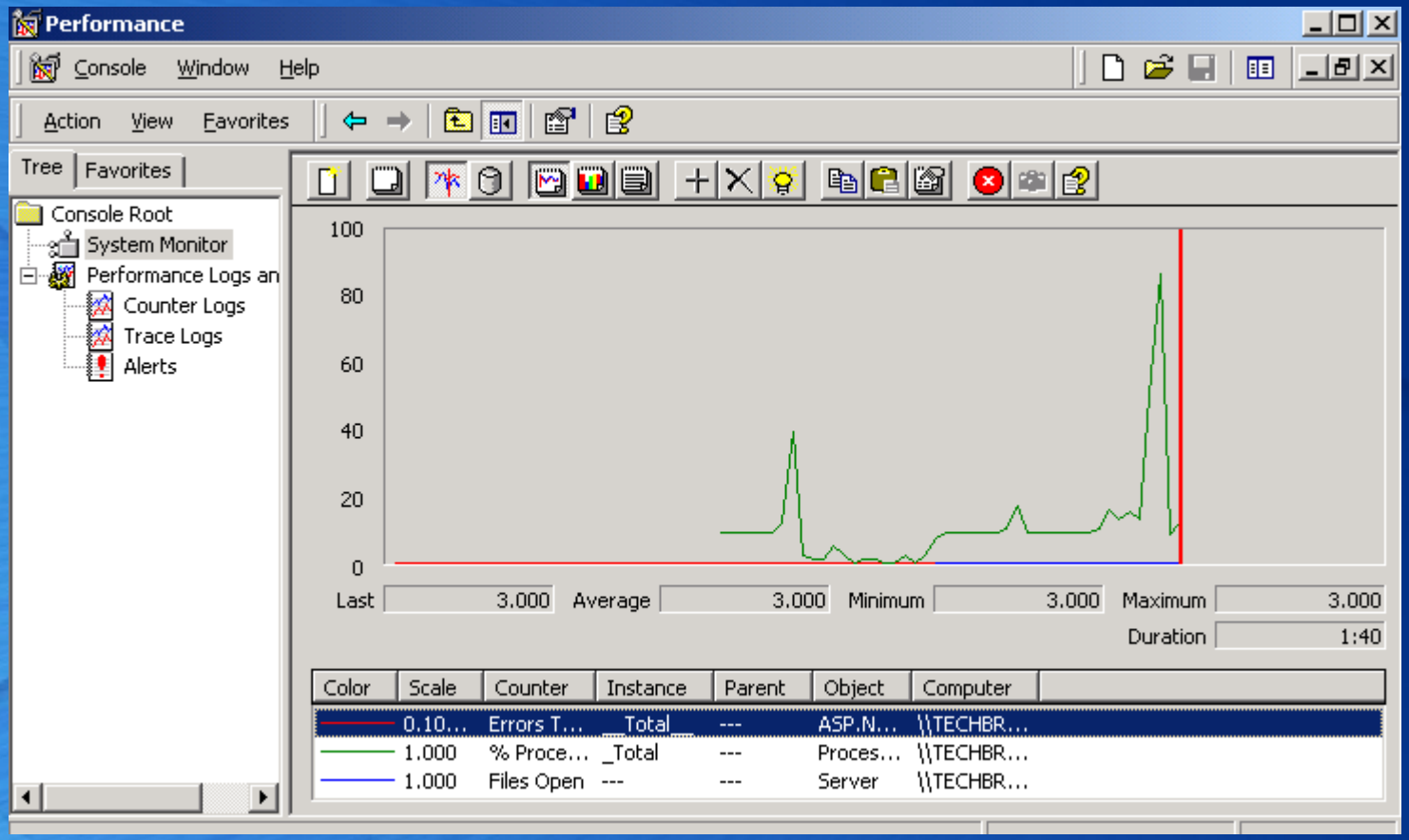
- Performance Counters
- Event Log
- Tracing

Application Diagnostics
Performance Counters

- Set and monitor custom or system performance counters
- Visualize using performance monitor MMC
- Use in Visual Studio .NET to create new categories and counters
  - Each counter has
    - Name
    - Type (NumberOfItems32, etc)
    - Description
  - You can also create counters by using
    - PerformanceCounterCategory.Create

Application Diagnostics
Performance Counters: Demo

- **Performance Monitor**



Application Diagnostics
Event Log

- Read from and write to custom or system event logs
- EventLog Class
  - Supports reading and writing to logs
    - Custom logs or system logs
    - Use with Server Explorer or entirely in code

Application Diagnostics
Event Log: Demo

- Event Log
- Add support to application for event logging

Application Diagnostics
Tracing

- **Instrument your code**
- **Systems.Diagnostics** namespace includes **Trace** and **Debug** classes
- **Choose listeners**
  - **Console: Windows-based debugger, the Output window will show Debug and Trace messages**
  - **File: The System.Diagnostic namespace provides a TextWriterTraceListener**
  - **Event log: The System.Diagnostic namespace provides an EventLogTraceListener**

Application Diagnostics
Tracing Demo

- To understand the Tracing features of .NET

Summary

- **Code Instrumentation and Middle Tier Technologies**
  - **Adding monitoring features**
  - **Using tracing to automate debugging**
  - **Building scalable features into applications (such as using MSMQ)**

Resources

- ***Basics of scaling applications*** **- Simple article on ASP, ASP+ and .NET**
  http://msdn.microsoft.com/msdnmag/issues/0900/aspplus/aspplus.asp

In-Depth Training on Scalable Applications

Direct from Microsoft

- 2350A: Securing and Deploying Microsoft .NET Assemblies (Prerelease)

10.24